

Bilkent University CS 319 Object-Oriented Software Engineering

Summer 2017

Analysis Report Project Name: Hotel Reservation System

Group #1

Mehmet Ali Altuntaş 21401004

Damla Eda Bıçakcı 21402130

Murat Süerdem 21401107

İrem Yurdakul 21400299

**Table Of Contents**

**1.Introduction**

**2.Overview**

**3.Requirement Specification**

3.1. Functional Requirements

3.2. Non-Functional Requirements

3.3 Constraints

**4. System Model**

4.1. Use Case Model

4.2. Dynamic Models

4.2.1 Scenarios and Sequence Diagrams

4.2.2 Activity Diagram

4.2.2.1. Making Reservation

4.2.2.2. Checking Room Feature

4.2.2.3. Deleting Reservation

4.3. Object and Class Model

**5. User Interface**

5.1 Finding Proper Room

5.2. Not Finding The Proper Room

5.3. Finding The Proper Room

5.4. Details About Room Seeing By User

5.5. Manager Main Page

5.6. View Room Details

5.7. Reservations For A Specific Room

**6.Conclusion**

**7.References**

**1.INTRODUCTION**

In today’s intense world, sometimes it may be hard to understand that the hotel you have been interested in is whether fully suitable for you or not. The most important factors while adjusting your hotel is related to the hotel’s joy and demand satisfying facilities. Our project has make possible to stay customers from every budget possible. In the way that our program do is after choosing number of people who are going to accommodate can increase the price for a specific room. If you choose just to stay in the room with basic form of it you pay less, otherwise with additional features like minibar, restaurant, housekeeping, laundry etc. you can stay in a luxury form. This property makes our program better, because for a good and peaceful holiday you should not be regret from the money that you have paid. In many booking system you can only choose your room with fix features, it becomes hard to change the money that you are going to pay, if it is not friendly with your budget you even have to change the hotel you choose. In addition, because we believe that the simpler of the program is more easy to understand, we specially focus on the plainness of the program. Therefore, we want to create a hotel reservation system that allows user to make their booking with pleasure and easy way.

The report structure is the following: Section 2 describes the case description. Section 3 includes the functional, non-functional requirements and constraints of the system. Section 4 covers the Use Case models and their descriptions. User interface is shown in Section 5. Section 6 shows class diagrams, dynamic models of the system which are state chart and sequence diagram with scenarios.

**2.OVERVIEW**

The system has three basic operation, which are make booking, specification of room features and cancel booking. These operations can be used by customer, who wants to stay in the hotel and also receptionist working for the hotel. In the hotel, there are 7 floors and at the each floor there are 20 rooms. Single rooms are in the first floor, double rooms are in the second floor so on. And also, odd numbered rooms have sea view and even numbered rooms have forest view.

Make Booking Part: When the user makes booking, first of all selects the check in and out dates and numbers of rooms before checking the number of people in the room/rooms. Therefore, user can make booking for multiple rooms in the same date range.

If there is an available room that has specified features, program passes the next step and can continue to pay the price or make the reservations for specified rooms.

If there is no available room, the user see a warning message that states desired room is not possible to make the reservation and direct his/her to the main page.

Specifications of Room Features Part: As mentioned before, users can add some room features or other type of services by themselves and according to this process, new price is going to be computed. However, just user’s demand is not going to be enough to make changes in room features, the receptionists should also give an approval to that request.

Cancel Booking Part: Customer has a right to cancel his/her booking. Additionally, receptionist can cancel a booking instead of customer or by the way of moving all the reservations for a specific room to another one, the receptionist can cancel all the reservations for a specific room. And lastly, the receptionist can cancel all the reservations in the hotel.

**3. REQUIREMENT SPECIFICATION**

**3.1. Functional Requirements**

In this system, there two types of users, which are customer and receptionist. They have different roles and authorisations on this system.

***For Customer:***

•   The customer should be able to select check in/out dates and number of people in rooms.

•   The customer should be able to make booking at most three rooms for the same date range.

•   The customer should be able to informed about rooms in hotel and select a room or rooms.

•   The customer should be able to select features to his/her room.

•   The customer should be informed about his/her reservation.

•   The customer should be able to cancel his/her reservation.

•   The customer should be able to make his/her payment.

***For Receptionist:***

•   The receptionist should be able to make booking for a customer.

•   The receptionist should be able to check and give approval to a room’s features that determined by customer.

•   The receptionist should be able to view information about a specific customer and     also about all customers in the rooms.

•   The receptionist should be able to cancel a reservation.

•   The receptionist should be able to view all rooms.

•   The receptionist should be able to view all reservations made for a specific room.

•   The receptionist should be able to move all reservations of any room to another room that fits with it, and delete the old room’s all reservations.

•   The receptionist should be able to make payment for a reservation of a customer.

•   The receptionist should be able to move a certain reservation for a room to another room.

**3.2. Non-Functional Requirements**

•   The system should response quickly at each step (1 second), because the main purpose is to make reservation quickly and efficiently.

•   Customer should be informed about other choices.

For example, if there is no available double room with sea view, other alternatives can be shown to the customer like double room with forest view instead of a sea view.

•   Program should be learned quickly by customers and receptionist, and used easily.

•   Extendibility : In software engineering, reusability and extendibility are two importance concepts. We plan to make Hotel Reservation System suitable to be extended and re-used for future works.

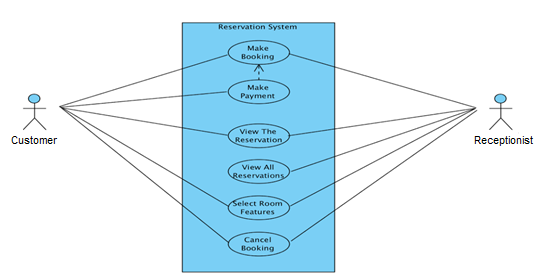
**3.3 Constraints**

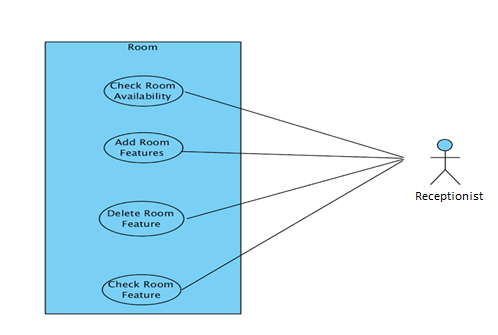
* The program must be a desktop application.
* The program must be implemented within six weeks.

**4. SYSTEM MODEL**

**4.1. Use Case Model**

This part provides information about use case model of Hotel Reservation System, detailed use case descriptions are included below.

****

****

**4.1.1**

***Use case name*** makeBooking

***Participating Actor*** Initiated by customer

***Entry condition*** Customer selects “make booking” button from main menu.

***Flow of Events***

1. Customer selects number of rooms to book

2. Customer selects check-in and check-out dates, number of people for each rom and rooms’ features.

3. Customer selects “show availabity of rooms and their prices”

button below the page.

4. System show the rooms information and prices.

5. If prices are suitable for the customer, he can selects

“confirm booking” button.

6. Customer can make payment by selecting “make payment” button, or

customer can make payment when he/she go to the hotel to host by

selecting “make payment in the hotel”.

***Exit condition*** System displays the reservation code.

**4.1.2**

***Use case name*** viewTheReservation

***Participating Actor*** customer and receptionist

***Entry condition*** Customer or receptionist selects “view the reservation” button from the main menu

***Flow of Events*** 1. Customer or receptionist write the reservation code.

***Exit condition*** Customer and receptionist see the reservation information(check-

in,check-out dates,room features,number of room etc.

**4.1.3**

***Use case name*** viewAllReservations

***Participating Actor*** Initiated by receptionist

***Entry condition*** Receptionist selects “view all reservation” button from main menu.

***Flow of Events*** 1. Receptionist enter the his/her password and selects “sign in” button.

2. Receptionist can choose any room button to see its all

reservations.

For example, if receptionist select the “105” button, he/she can

see the all reservations for room 105.

***Exit condition*** Receptionist selects “back to main menu” button.

**4.1.4**

***Use case name*** cancelBooking

***Participating Actor*** customer and receptionist

***Entry condition*** Customer or receptionist selects “cancel booking” button from main menu.

***Flow of Events*** 1. Customer or receptionist enter the reservation code

and selects “cancel” button.

***Exit condition*** System give information about whether canceling booking canceled succefully or not.

**4.1.5**

***Use case name*** addRoomFeatures

***Participating Actor*** Initiated by receptionist

***Entry condition*** Receptionist selects “add room feature” button from main menu.

***Flow of Events*** 1. Receptionist enter the his/her password and selects “sign in” button.

2. Receptionist selects the room to activate any feature of it.

3. Receptionist tick up to add feature to specified room by

activating it.

***Exit condition*** Receptionist selects “ save changes” button and system give information about whether changes applied succesfully or not.

**4.1.6**

***Use case name*** deleteRoomFeatures

***Participating Actor*** Initiated by receptionist

***Entry condition*** Receptionist selects “delete room feature” button from main menu.

***Flow of Events*** 1. Receptionist enter the his/her password and selects “sign in” button.

2. Receptionist selects the room to inactivate any feature of it.

3. Receptionist remove tick to delete feature of specified room by

inactivating it.

***Exit condition*** Receptionist selects “ save changes” button and system give information about whether changes applied succesfully or not.

**4.1.7**

***Use case name*** checkRoomFeatures

***Participating Actor*** Initiated by receptionist

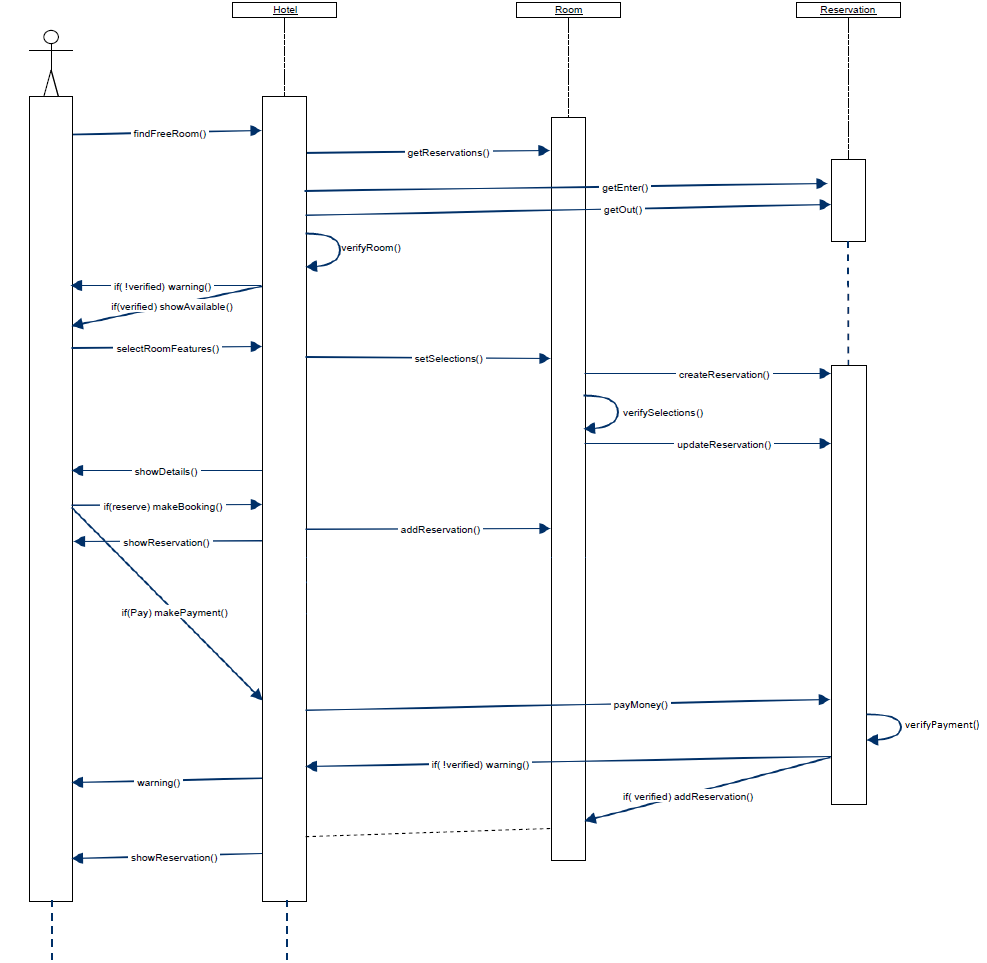
***Entry condition*** Receptionist selects “check room feature” button from main menu.

***Flow of Events*** 1. Receptionist enter the his/her password and selects “sign in” button.

***Exit condition*** Receptionist selects the room to see any feature of it.

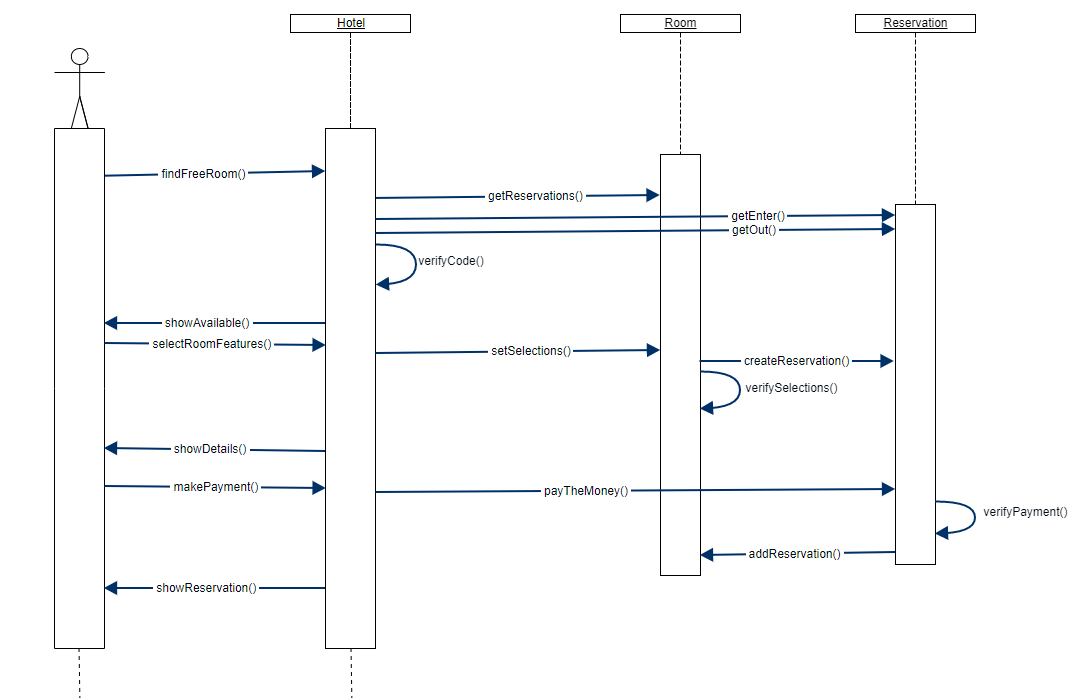
**4.2.1 Scenarios and Sequence Diagrams**

**4.2.1.1 Scenario Name**: Make booking and payment

**Scenario**: Ahmet wants to make a booking for himself and his friends for multiple rooms. He starts to use the system first of all, he selects the check in/out dates and number of rooms. After all selections, have been done, if there is not any available room, he will see a warning message related to its unavailability. Otherwise, Ahmet can select the room features that he wants and then system sets the selections for the room and updates the reservation details and shows it. Ahmet can either make the reservation and later pay the fee or he can both pay the fee and make the reservation. If he chooses to make booking, the reservation is added and shown to him, otherwise, Ahmet pays the fee and then reservation is shown to Ahmet. If he does not want to reserve the room he can turn back to main menu.

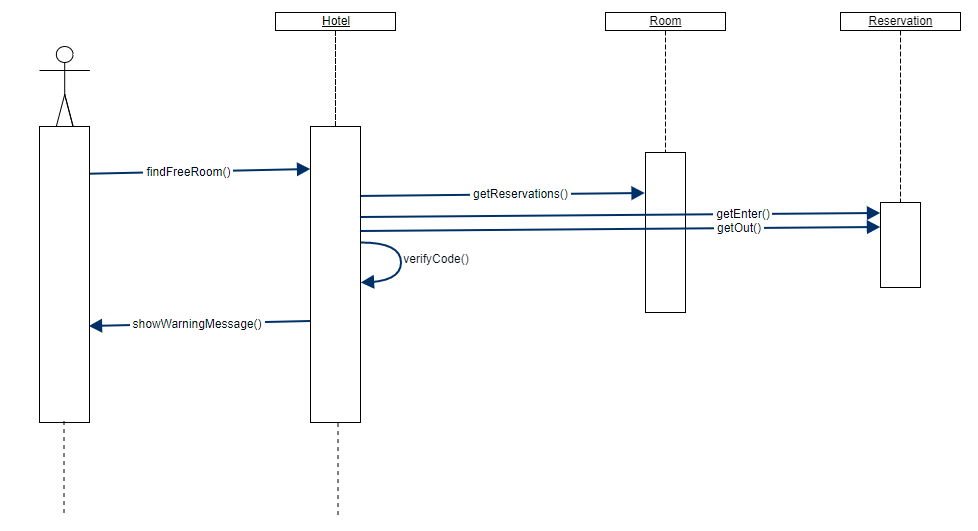
**4.2.1.2 Scenario Name**: Make booking

**Scenario**: Ahmet wants to make a booking for himself and his friends for multiple rooms. He starts to use the system, first of all he selects the check in/out dates and number of rooms and types of rooms to determine the specifications. After all selections, have been done if he is comfortable with the price he can make the payment. After all the operations, he is going to be informed about his reservations and rooms’ features, and make booking is finished.

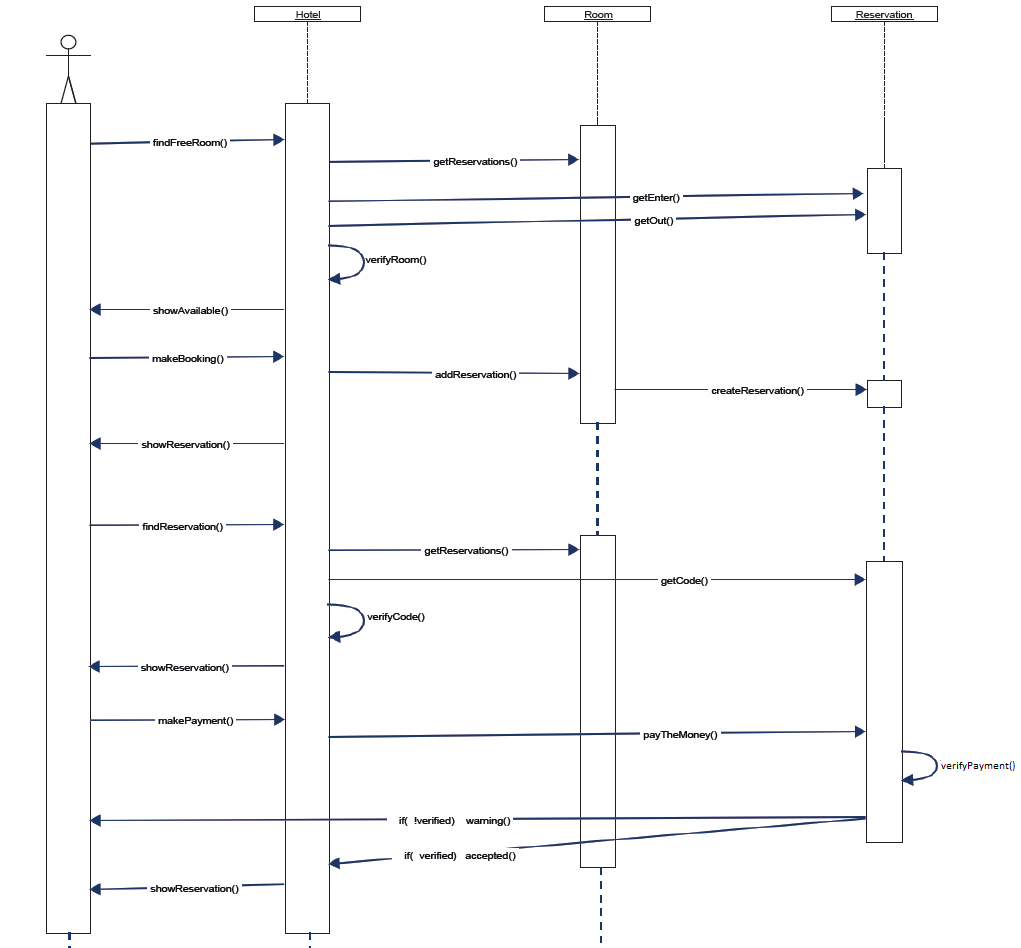


**4.2.1.3 Scenario Name**: Make booking(Negative)

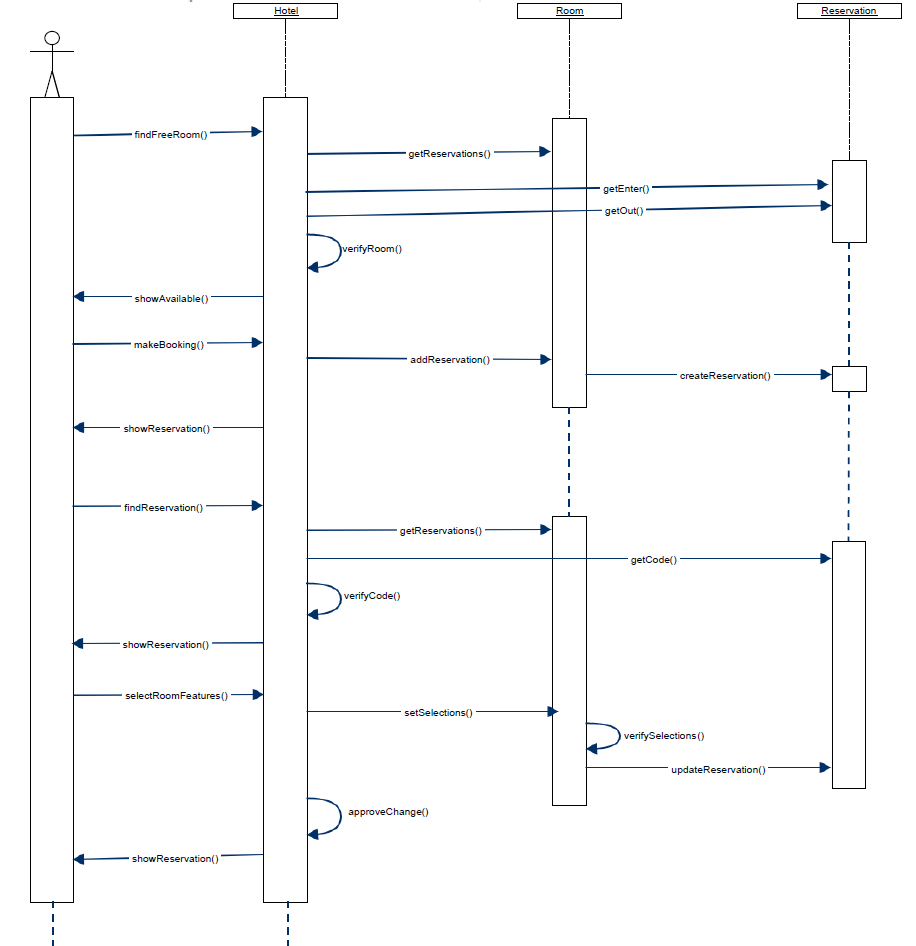
**Scenario**: Can wants to make a booking for himself and his friends for multiple rooms. He starts to use the system first of all, he selects the check in/out dates and number of rooms and types of rooms to determine the specifications. After all selections, have been done, he realizes that there is no available room for his desired type. Therefore, he is informed by the system about this situation.



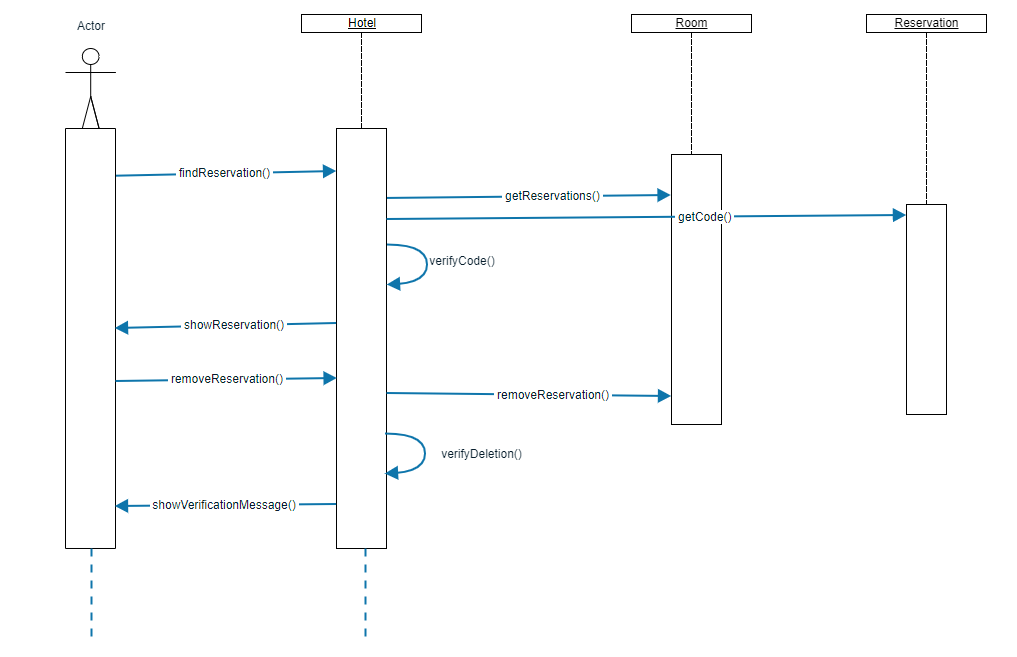
**4.2.1.4 Scenario Name:** Make payment later

**Scenario:** Mehmet came from the work very tired and decided to go on a vacation. After making the reservation via the system, he also wants to the make the payment. However, his credit card limit is not enough to pay the fee so, he reserves the available room. Later then, he enters the system and searches for the reservation by the reservation code and pays the fee.

**4.2.1.5 Scenario Name:** Add specifications after booking

**Scenario:** Murat wants to make booking for a double room. He selects check in/out dates and number of people in the room, and then he chooses an available room from the hotel and reserves the room. However, he decides to add new feature to his room. He selects the mini-bar option and cancel the housekeeping service, then his reservation is approved by the receptionist, and booking is finished, and Murat is informed about his reservation.

**4.2.1.6 Scenario Name**: Cancel booking

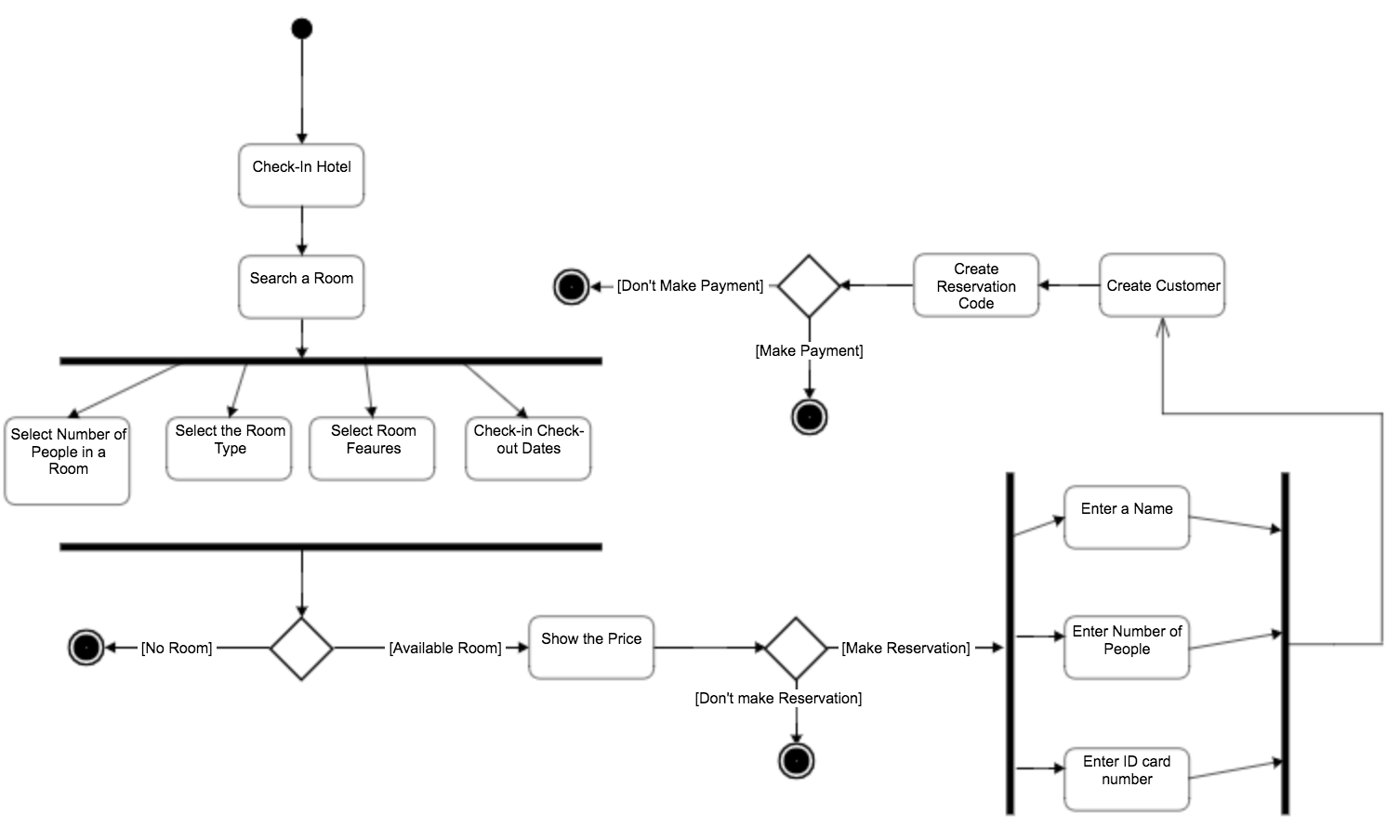
**Scenario:** Eda was made booking in hotel, but she cannot go holiday in specified dates because of her job. Therefore, she enters the system and cancels her current booking.

**4.2.2. Activity Diagram**

This part includes diagrams and explanations about making reservation by user and changing room features of a room by the manager.

**4.2.2.1. Making Reservation**

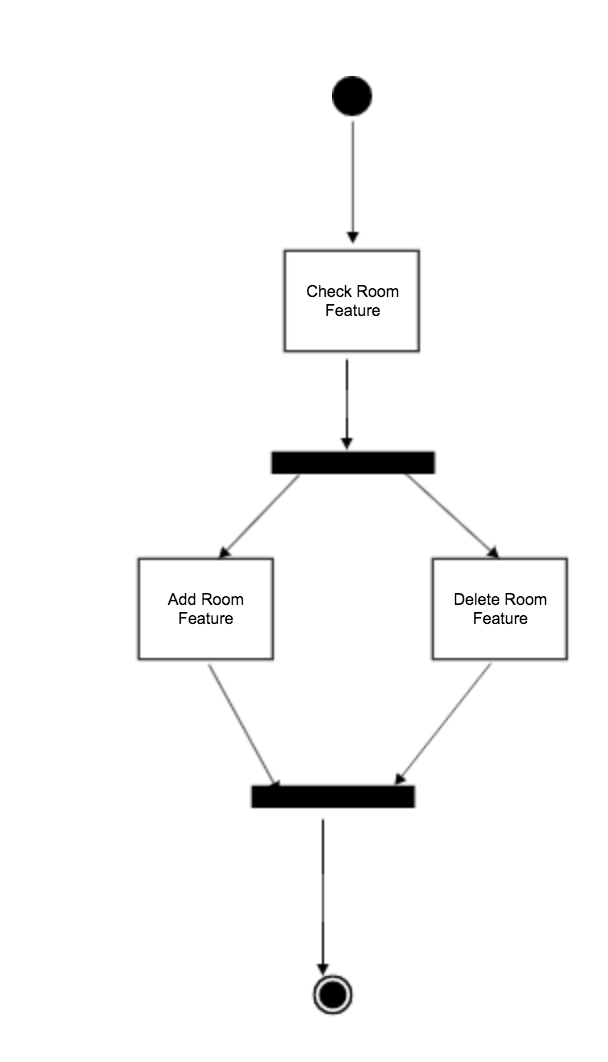
Following activity diagram shows how user complete the taking a reservation to the system.

****

When the customer has entered the reservation system first of all he should find the most suitable room to himself. To search for a room, he should be fulfilled the information about number of people that are going to accommodate in the hotel, type of the room according the room view, choosing optional room services like laundry or housekeeping and check-in and check-out dates that the customer going to stay simultaneously. These four selections can be done without a selection order. After that room selection if there is no suitable room for customer the program ends here the maintaining making the reservation operation, unless the program shows the price of the room and starts to collect customer information. It takes the name, last name and citizenship number and finalise collecting customer information. Then program creates a reservation code. At the end of that process customer can choose to pay for the room or not to pay currently, in both ways customer ends the making reservation.

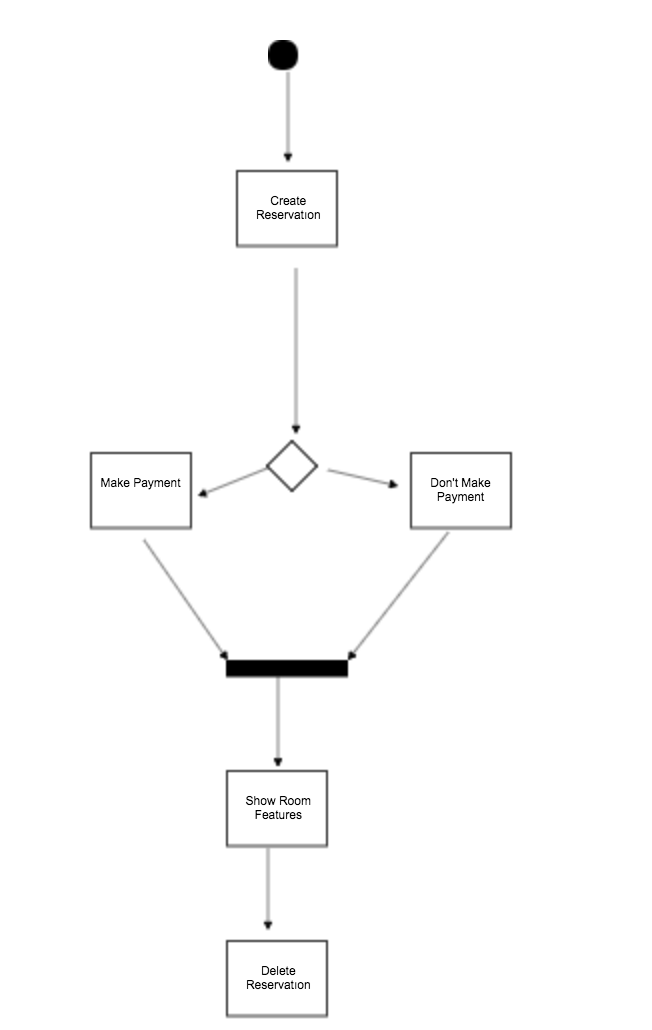
**4.2.2.2. Checking Room Feature**

Following activity diagram shows how manager complete the adding and deleting of the room features to the system.

****

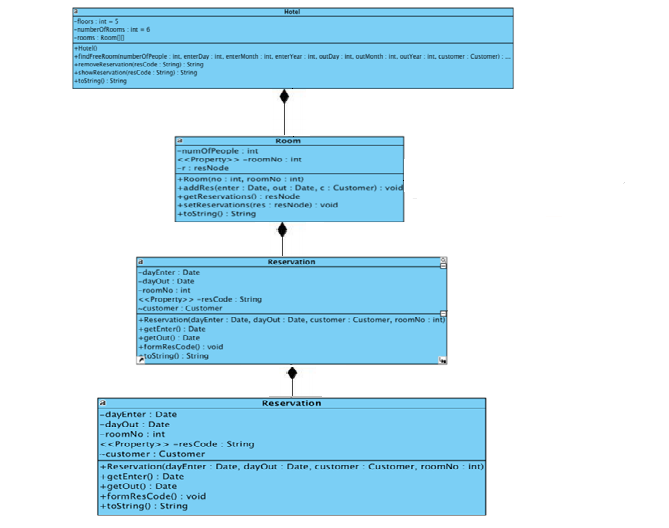
When the manager enters the system to make an addition or deletion from/to the room features, first he checks for the room features. Then he can both adding or deleting a room feature from/to the system simultaneously. And then completes the process.

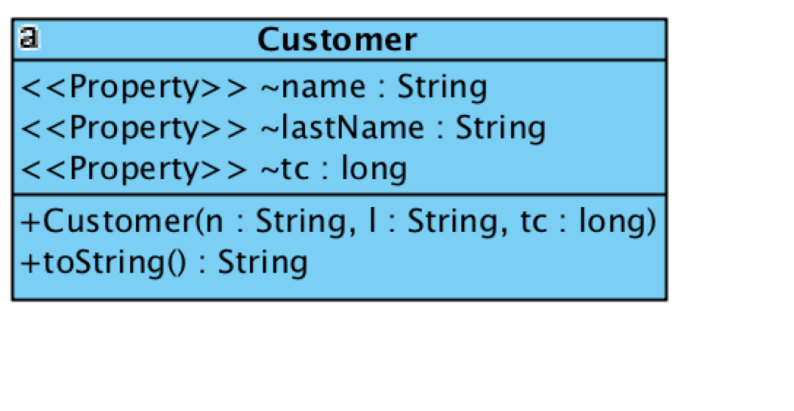
**4.2.2.3. Deleting Reservation**

****

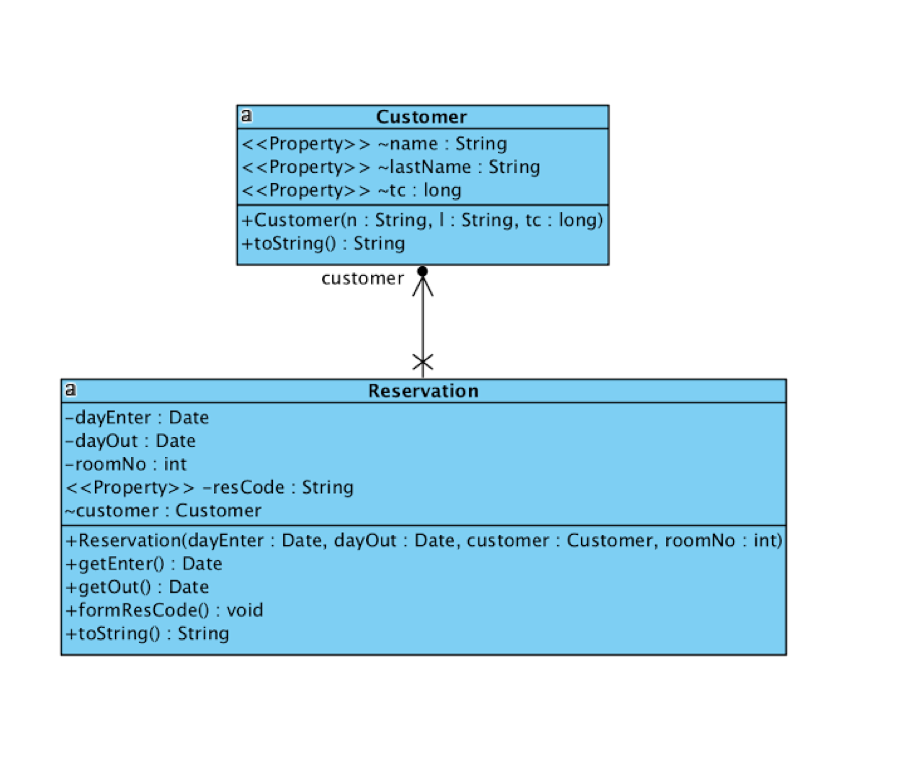
When the user has created its reservation. He can make the payment immediately or choose not to do. In both cases the program direct the user a page that shows the room features. After all the creation process has completed the user can delete the reservation from the system.

**4.3. Object and Class Model**

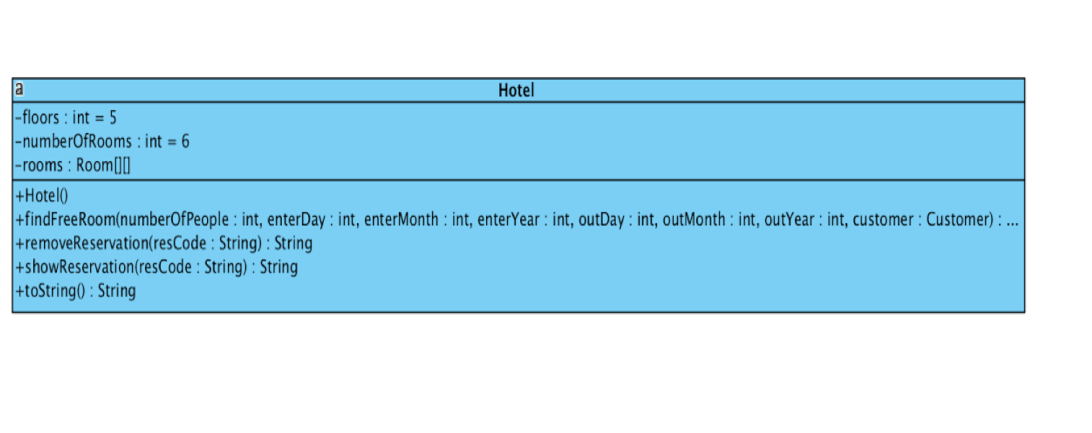
****

****

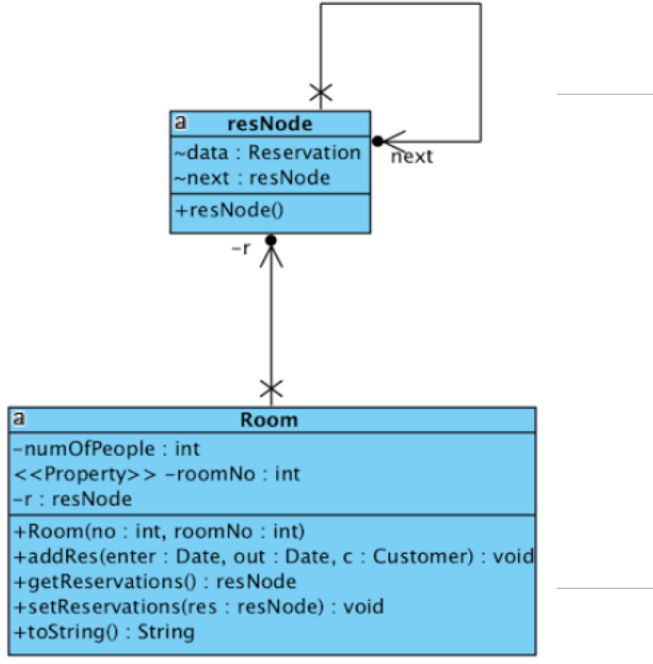
Customer Class, is a class that represents customer object. That has name, lastName, and Id number. Every reservation object has a customer who give her name, lastName, and Id number while making reservation.

****

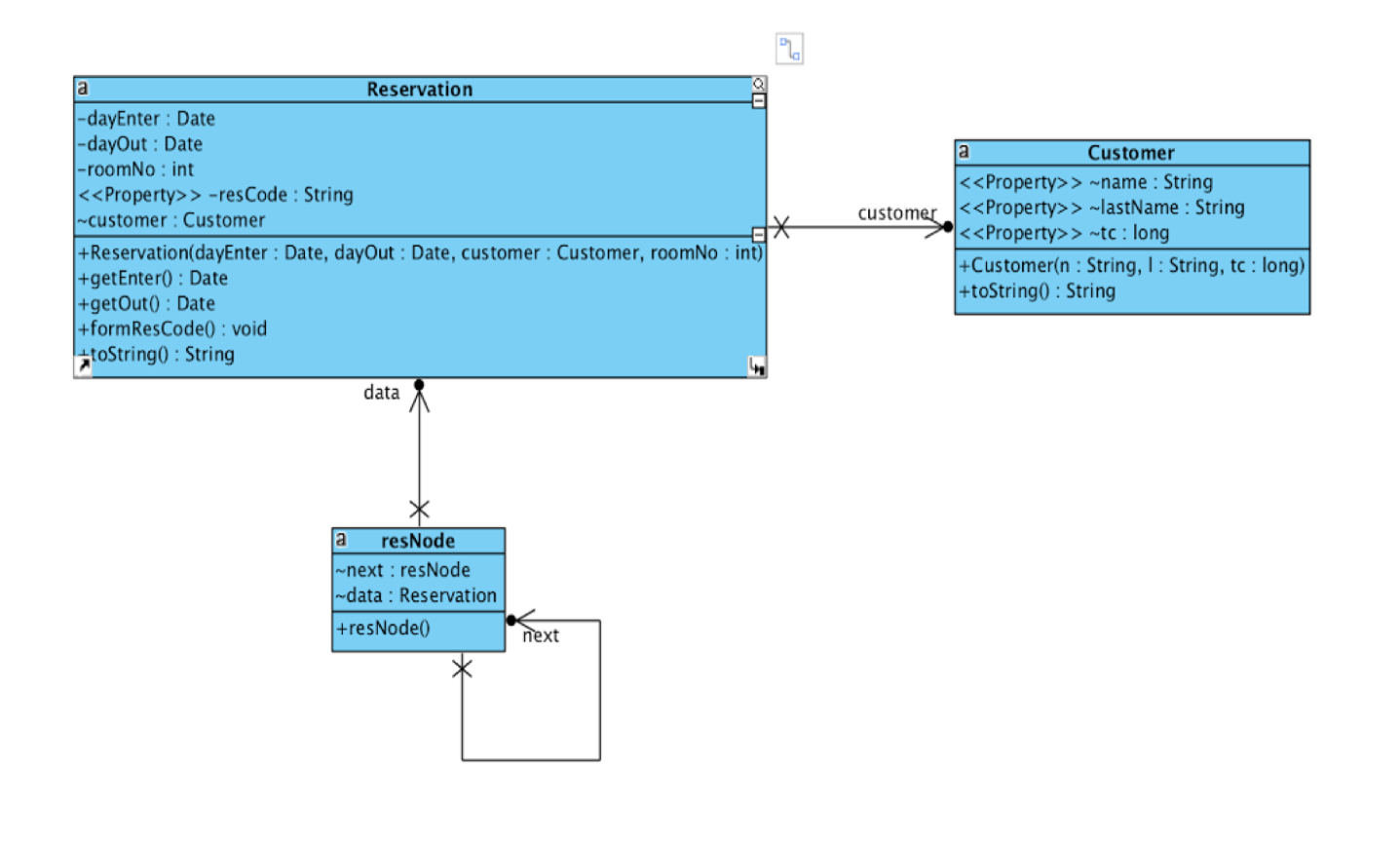
Reservation class, is a class that has check in/out dates and room number basically. It performs to create a reservation code for a reservation.

****

Hotel class, is a class that has number of rooms and floors. Its functionalities are findingavailable room, removing a reservation, and showing a reservation.

****

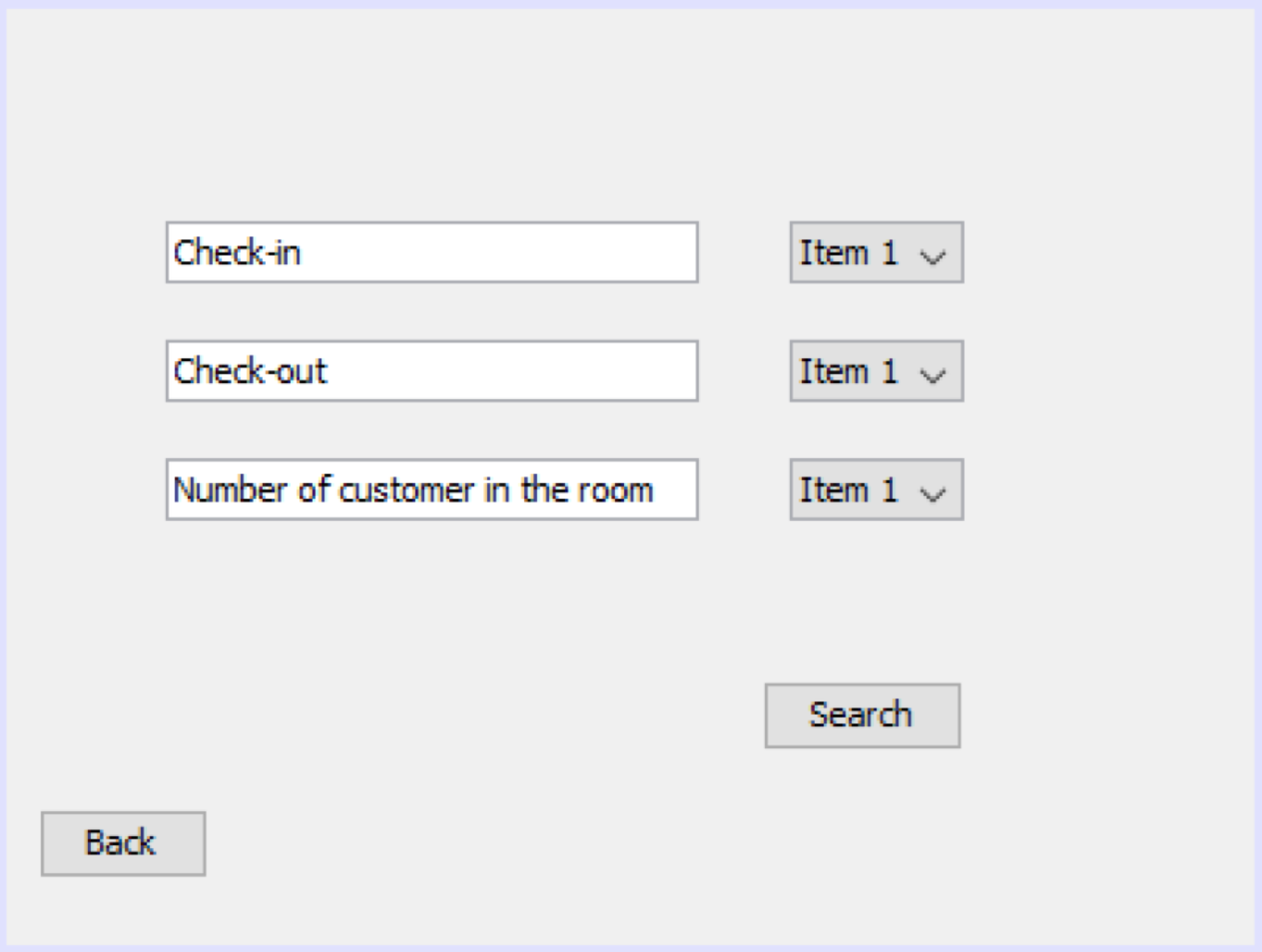
Room class, is a class that has number of people in it. It keep all reservations for every specified rooms. It is responsible for adding and setting reservations that has.

****

This diagram shows the relations between classes.

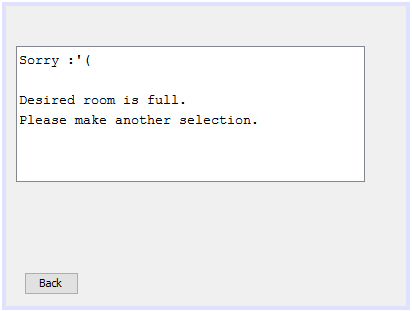
**5. USER INTERFACE**

**5.1 Finding Proper Room**

****

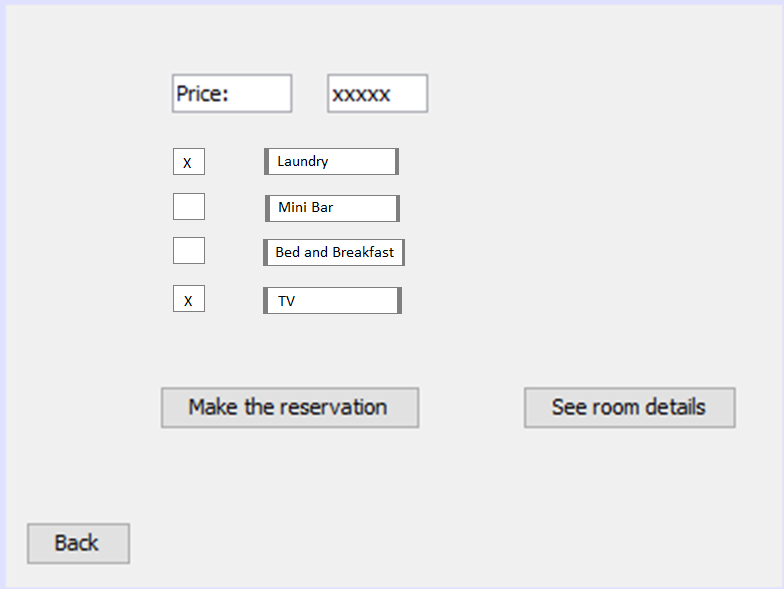
When the user first enters the system for finding a proper room according to his requests, He can select proper dates for check-in and check-outs and number of customers that are going to be accommodate in the room from the combo boxes on the left.

**5.2. Not Finding The Proper Room**

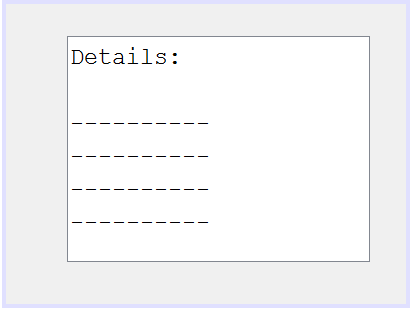
****

When user has clicked the search button after selecting desired properties, if there is no available room for user will see that message which states the unavailability of the room on the screen.

**5.3. Finding The Proper Room**

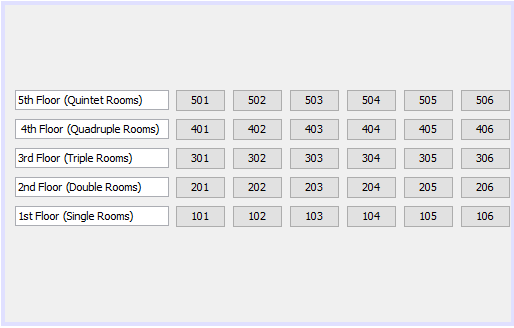
****

When user has clicked the search button after selecting desired properties; if there is an available room for user, there is going to be a page appears stating the price of the room. If user wants optional room features, he can select them from the combo boxes. If user want to see the details about the room he can basically click the “See room details” button. If the user comfortable with the price he can basically click the “Make the reservation” button and complete the process.

**5.4. Details About Room Seeing By User**

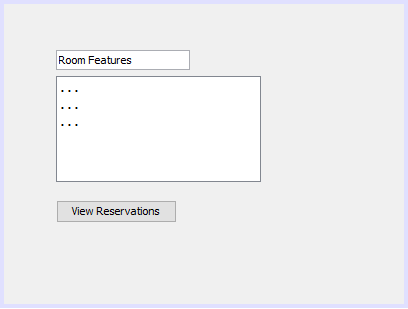
The page above will appear when the user wants to see the details about the room that he has already choose of its features.

**5.5. Manager Main Page**

****

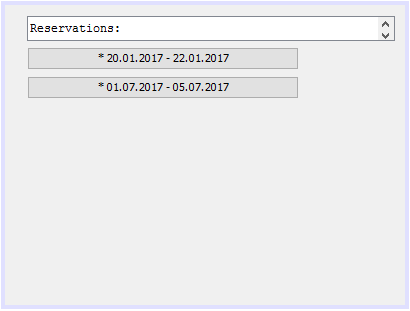
When the user first enters the system he will see the page above. The left side states floor number with typical room feature and every number from 101 to 506 hold information about the reservations of the room. If the manager wants to see the reservations and room features in detail, he can basically click the room number he can see all those information gradually.

**5.6. View Room Details**

****

After the manager clicked desired number he saw the room features of desired room number. If the manager continues to see the all reservations saved in the system, basically he can click the “View Reservation” button.

**5.7. Reservations For A Specific Room**

****

After the manager has clicked a room number that is wanted to be checked for features, the manager will see the page above. The page list all the reservations for the desired room, if the manager wants to see the details about any reservation basically he can click the dates.

**6.CONCLUSİON**

In this report, our aim is to analyse our project, which is Hotel Reservation System for design and implementation parts. First of all, we focus on requirements part and second part is about system model.

In the requirements part, we stated functional requirements which are essential for our project, and non-functional requirements that are also important for our project. Our aim is to provide all requirements for a better project. We examined requirements for each actor in the system. Also, at this point we implemented a few codes for prototyping to obtain better understanding in our project.

In the second part, which is system model we started with use case models, these use cases are very useful to highlight the basic functionalities of our project. We examined actors, functionalities of our system and relations between actors and the system functions. After that, we created some possible scenarios to understand the performance of our system. And we created sequence diagrams which demonstrate sequence between functions and objects for each scenario. And we used activity diagrams to highlight the main activities and functionalities in our system.

By using these collection of information, our aim is to design a good system and good implementing.

**7.REFERENCES**

* [**https://www.booking.com/index.tr.html?label=gen173nr-1BCAEoggJCAlhYSDNiBW5vcmVmaOQBiAEBmAEowgEKd2luZG93cyAxMMgBDNgBAegBAZICAXmoAgM;sid=1b8b2da1dc69166a1238f0acf647fe1b;sb\_price\_type=total&**](https://www.booking.com/index.tr.html?label=gen173nr-1BCAEoggJCAlhYSDNiBW5vcmVmaOQBiAEBmAEowgEKd2luZG93cyAxMMgBDNgBAegBAZICAXmoAgM;sid=1b8b2da1dc69166a1238f0acf647fe1b;sb_price_type=total&)
* [**https://www.gliffy.com/**](https://www.gliffy.com/)